

Politechnika Warszawska

W Y D Z I A Ł M A T E M A T Y K I
I N A U K I N F O R M A C Y J N Y C H



Praca dyplomowa inżynierska

na kierunku Informatyka

Interfejs użytkownika do manualnego obrysu struktur oraz wybranych
anomalności w obrazach medycznych z wykorzystaniem tabletu
graficznego

Łukasz Garstecki

Numer albumu 276857

Tomasz Świerczewski

Numer albumu 276915

promotor

dr inż. Magdalena Jasionowska

WARSZAWA 2019

.....

podpis promotora

.....

podpisy autorów

Streszczenie

Interfejs użytkownika do manualnego obrysu struktur oraz wybranych anormalności w obrazach medycznych z wykorzystaniem tabletu graficznego

Przykładowe streszczenie. Do wykonania jako ostatnie.

Słowa kluczowe: slowo1, slowo2, ...

Abstract

English title

Sample abstract in english.

Keywords: keyword1, keyword2, ...

Warszawa, dnia

Oświadczenie

Oświadczam, że moją część pracy inżynierskiej (zgodnie z podziałem zadań opisanym na wstępie) pod tytułem „Interfejs użytkownika do manualnego obrysu struktur oraz wybranych anormalności w obrazach medycznych z wykorzystaniem tabletu graficznego”, której promotorem jest dr inż. Magdalena Jasionowska, wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....

Spis treści

Wstęp	12
1 Wprowadzenie	13
1.1 Zagadnienia medyczne związane z aplikacją	13
1.2 Podział prac	13
2 Stan wiedzy	14
2.1 Przegląd istniejących rozwiązań	14
2.2 Proponowane rozwiązanie	14
3 Opis autorskiego systemu informatycznego	15
3.1 Specyfikacja wymagań	15
3.1.1 Opis biznesowy	15
3.1.2 Wymagania funkcjonalne	15
3.1.3 Wymagania нефункционалне	17
3.2 Architektura rozwiązania	17
3.3 Opracowany algorytm półautomatyczny	17
3.3.1 Wykrycie krawędzi na bitmapie	19
3.3.2 Stworzenie grafu z bitmapy	21
3.3.3 Zapewnienie spójności grafu	24
3.3.4 Wyszukanie najkrótszych ścieżek w grafie	24
3.3.5 Optymalizacja	24
3.4 Moduł obliczeń statystyk	24
4 Przeprowadzone eksperymenty	25
4.1 Zbiór testowy	25
4.2 Wydajność algorytmu półautomatycznego	25
4.3 Analiza wyników i wnioski	25

5 Podsumowanie	26
5.1 Napotkane problemy i ograniczenia	26
5.2 Możliwości dalszego rozwoju	26
Bibliografia	27
Instrukcja instalacji	29
Instrukcja użytkowania	30
Wykaz symboli i skrótów	31
Spis zawartości załączonej płyty CD	32

Wstęp

O czym jest praca? Co się w niej znajduje? Jaki jest wkład autora?

1. Wprowadzenie

1.1. Zagdanienia medyczne związane z aplikacją

1.2. Podział prac

2. Stan wiedzy

2.1. Przegląd istniejących rozwiązań

2.2. Proponowane rozwiązanie

3. Opis autorskiego systemu informatycznego

W poniższym rozdziale zawarto dokumentację techniczną i biznesową tworzonego systemu. Przedstawiono w szczególności: wymagania, architekturę, zastosowane metody półautomatycznego obrysu oraz metody obliczania statystyk obrysu.

3.1. Specyfikacja wymagań

3.1.1. Opis biznesowy

Celem projektu jest stworzenie interfejsu przyjaznego użytkownikowi, który umożliwi przeglądanie plików DICOM, a także przeprowadzanie na tych plikach obrysów. Prace obejmują stworzenie aplikacji webowej, która udostępni użytkownikowi interfejs komunikujący się z bazą danych Orthanc oraz serwera odpowiedzialnego za przechowywanie wygenerowanych przez użytkownika obrysów oraz wyznaczanie obrysów półautomatycznych.

Do podstawowych funkcjonalności systemu zaliczają się:

- Generowanie obrysu manualnego.
- Generowanie obrysu półautomatycznego na podstawie punktów podanych przez użytkownika.
- Zapisywanie wygenerowanych obrysów.
- **Anonimizacja**¹ danych zapisanych w strukturze pliku DICOM.

3.1.2. Wymagania funkcjonalne

Poniżej przedstawiono wymagania funkcjonalne w postaci historyjek użytkownika (ang. user stories):

¹ Anonimizacja (ang. anonymization) — operacja mająca na celu usunięcie z danych informacji o pacjentach, które pozwoliłyby na identyfikację danych z tożsamością pacjenta. Są to między innymi: imiona, nazwisko, pesel. Inne tłumaczenia słowa anonymization — utajnianie, usuwanie danych niejawnych. Z uwagę na fakt, że te tłumaczenia nie oddają dobrze kontekstu zastosowano kalkę językową.

1. Jako użytkownik chcę wczytać obraz DICOM.

Użytkownik może wybrać obraz w menu bocznym, w którym ma możliwość wyboru pacjenta, badania oraz serii. Wybranie serii skutkuje wyświetleniem pierwszego obrazu DICOM z tej serii.

2. Jako użytkownik chcę zmienić obraz w serii przy użyciu rolki myszy.

Po najechaniu na obraz przewijanie rolką myszy do góry powoduje zmianę wyświetlanego obrazu na kolejny obraz w serii. Gdy przewijamy rolką myszy do góry na ostatnim obrazie w serii wyświetlany obraz nie zmienia się. Analogicznie przewijanie rolką myszy w dół powoduje zmianę wyświetlanego obrazu na poprzedni obraz w serii, a przewijanie w dół rolką myszy na pierwszym obrazie w serii nie powoduje zmiany obrazu.

3. Jako użytkownik chcę wykonać obrys przy użyciu tabletu graficznego.

Po najechaniu na obraz kursorem sterowanym przez tablet graficzny, po wciśnięciu końcówki rysika użytkownik prowadzi kursor po obrazie wykonując obrys bez odrywania końcówki rysika od podkładki. Jeżeli użytkownik nie zakończy obrysu dokładnie w punkcie, w którym go rozpoczął, obrys powinien zakończyć się linią prostą, łączącą punkt końcowy z punktem początkowym.

4. Jako użytkownik chcę wygenerować obrys na podstawie wybranych punktów.

Po najechaniu kursorem na obraz użytkownik może wybierać punkty, na podstawie których zostanie wygenerowany obrys, poprzez wciśnięcie lewego przycisku myszy w miejscach, w których chce, aby znalazły się punkty. Użytkownik może zobaczyć efekt wygenerowanego przez system obrysu

5. Jako użytkownik chcę edytować listę punktów, z której wygenerowany zostanie obrys.

Użytkownik może usunąć wcześniej wybrany punkt po najechaniu na niego kursorem i wciśnięciu lewego przycisku myszy. Użytkownik może dodać nowy punkt do listy punktów poprzez wciśnięcie lewego przycisku myszy w miejscu, w którym chce wstawić punkt.

6. Jako użytkownik chcę wybrać kolor obrysu.

Użytkownik wybiera kolor z palety kolorów lub zdefiniować własny kolor poprzez podanie numeru RGB koloru, który chce wybrać.

7. Jako użytkownik chcę zapisać obrys.

Po wykonaniu obrysu manualnego lub wybraniu listy punktów do wygenerowania obrysu półautomatycznego użytkownik wybiera nazwę obrysu i zapisuje obrys w systemie.

8. Jako użytkownik chcę obejrzeć zapisany obrys.

Użytkownik wybiera z listy po prawej stronie zapisany obrys i przegląda obrys naniesiony na obraz, na którym został wykonany.

9. Jako użytkownik chcę zobaczyć statystyki dotyczące obrysu.

Użytkownik wybiera z listy po prawej stronie zapisany obrys i przegląda statystyki obliczone na podstawie zapisanego obrysu. Do statystyk zalicza się obwód obrysu, pole obrysu, histogram obrazu na obszarze obrysu oraz liczba pikseli wewnątrz obrysu.

10. Jako użytkownik chcę zobaczyć jednocześnie dowolną liczbę zapisanych w systemie obrysów na jednym obrazie DICOM.

Użytkownik wybiera poprzez kliknięcie lewym przyciskiem myszy na nazwie obrysu znajdującej się na liście po prawej stronie. Wybrane obrysy wyświetlane są jednocześnie na przeglądającym przez użytkownika zdjęciu. Użytkownik może wyłączyć podgląd wcześniej wybranego obrysu poprzez ponowne wciśnięcie lewego przycisku myszy na nazwie obrysu na liście po prawej stronie. Na zdjęciu wyświetlane są jedynie obrysy wykonane na tym obrazie.

11. Jako użytkownik chcę zanonimizować dane pacjenta zawarte w pliku DICOM.

Użytkownik może zanonimizować pacjenta, gdy przegląda jego obraz. Użytkownik może anonimizować imię i nazwisko pacjenta, datę urodzenia pacjenta oraz płeć pacjenta poprzez nadanie nowych wartości lub poprzez usunięcie poprzedniej wartości i pozostawienie pustych pól w formularzu.

3.1.3. Wymagania niefunkcjonalne

Tabela 3.1 przedstawia wymagania niefunkcjonalne, które system musi spełnić.

3.2. Architektura rozwiązania

3.3. Opracowany algorytm półautomatyczny

Opracowany algorytm półautomatyczny służy do wykrywania krawędzi na obrazie medycznym. Jest algorytmem półautomatycznym, ponieważ jest wspomagany przez człowieka — użytkownika, który wybiera punkty na ekranie. Te punkty są interpolowane przez algorytm półautomatyczny, zwany dalej algorytmem.

Jako dane wejściowe do algorytmu uzyskujemy następujące informacje:

Tablica 3.1: Spis wymagań niefunkcjonalnych

Obszar wymagań	Nr wymaga- nia	Opis
Użyteczność (ang. Usability)	1	Każda funkcjonalność aplikacji dostępna dla użytkownika musi mieścić się na pojedynczym ekranie przy rozdzielczości 1920x1080 i czcionce nie mniejszej niż 12pt.
	2	Aplikacja powinna udostępniać pobranie zapisanych obrysów przy użyciu serwisu REST.
Niezawodność (ang. Reliability)	3	Aplikacja ma być dostępna 24h w ciągu doby. Dopuszczalne jest brak działania aplikacji w dowolnym momencie przez okres nie dłuższy niż przez 12h. Po przerwie w działaniu aplikacja musi być dostępna przez kolejne 24h bez utrudnień.
Wydajność (ang. Performance)	4	Aplikacja powinna pobierać dane zewnętrzne w postaci pliku DICOM (około 20MB) nie dłużej niż 5 sekund
	5	Aplikacja powinna generować obrys półautomatyczny i zapisywać obrys do systemu w czasie nie dłuższym niż 30 sekund.
	6	Aplikacja powinna reagować na działanie użytkownika (z wyłączeniem generowania obrysu półautomatycznego i zapisu obrysu do systemu) w czasie nie dłuższym niż 1 sekunda.

3.3. OPRACOWANY ALGORYTM PÓŁAUTOMATYCZNY

- Identyfikator obrazu medycznego, na którym był wykonywany obrys.
- Lista punktów wybranych przez użytkownika. Punkty te zostały wcześniej przeskalowane ze współrzędnych w aplikacji internetowej (aplikacji webowej, ang. web application) na współrzędne odpowiadające rozdzielczości obrazu medycznego.

Algorytm można podzielić na kilka ważnych etapów:

- Wykrycie krawędzi na bitmapie,
- Stworzenie grafu z bitmapy,
- Zapewnienie spójności grafu,
- Wyszukanie najkrótszych ścieżek w grafie.

Poniżej zostaną przedstawione dokładne rozwiązania dla każdego z tych kroków. Przed rozpoczęciem przetwarzania jest pobierany obraz medyczny o danym wcześniej identyfikatorze ze serwera Orthanc. Jest on podstawą do dalszej pracy algorytmu.

3.3.1. Wykrycie krawędzi na bitmapie

Często na obrazach medycznych różnice w charakterystyce poziomów szarości pikseli reprezentujących interesujące nas obiekty są małe, nie są dane dodatkowe informacje o naturze obrazu. Problem opracowania uniwersalnego algorytmu wykrywania krawędzi jest problemem trudnym. Dla wielu algorytmów można znaleźć takie przykłady, że te algorytmy nie wyznaczają poprawnie krawędzi. Ponadto są wymagania dla stwierdzania poprawności działania danego algorytmu, czy też operatora morfologicznego. Zgodnie z [4] „dobry detektor krawędzi powinien spełniać następujące warunki:

- niskie prawdopodobieństwo zaznaczenia punktów nienależących do krawędzi oraz niskie prawdopodobieństwo niezaznaczenia punktów należących do krawędzi,
- zaznaczone punkty krawędzi powinny być możliwie blisko jej osi,
- wyłącznie jedna odpowiedź na pojedynczy punkt krawędzi.”

Po zapoznaniu się z literaturą związaną z przetwarzaniem obrazów medycznych w algorytmie został użyty operator Canny’ego. Jest on powszechnym i dobrze sprawdzonym rozwiązaniem do wykrywania krawędzi. Jak napisał autor [4] „Operator ten (Canny’ego) jest bardzo popularny, chętnie wykorzystywany i adoptowany do wielu zastosowań. (...) Stał on się również standardem często używanym do porównań innych metod wykrywania krawędzi.” Zgodnie z rysunkiem 4.25 „Porównanie operatorów wykrywania krawędzi” w [4] najlepiej wykrywał główne narządy, takie jak wątroba czy też trzustka. Z wyżej wymienionych powodów został on wykorzystany w tym algorytmie.

Operator Canny'ego [5] składa się z 3 zasadniczych kroków:

1. Określenie wartości i kąta gradientu.

W tym celu został wykorzystany operator gradientu, a estymatorem gradientu w funkcji dyskretniej, jaką jest obraz zastosowano maskę, czy też operator Sobela. Wykorzystując go uzyskano dla każdego piksela wielkość oraz kierunek gradientu, co służy do dalszych obliczeń.

2. Wykrycie miejsc występowania krawędzi.

W tym celu został wykorzystany algorytm non-max suppression. Polega on na wyborze takich pikseli, które mają największą wartość gradientu na linii o kierunku zgodnym z kątem danego gradientu. Możliwe są 4 kierunki: pionowy, poziomy oraz dwa diagonalne. Jeśli dany piksel miał większą wartość gradientu od dwóch swoich sąsiadów, to zaznaczono go jako potencjalny punkt tworzący krawędzie. W ten sposób otrzymano obraz z potencjalnymi krawędziami.

3. Wyznaczanie krawędzi progowaniem histerezy

Po poprzednim kroku na obrazie nadal znajdują się nieistotne krawędzie. W tym celu Canny wprowadził ideę progowania histerezy. Metoda ta wymaga 2 wartości progowych T_1, T_2 takich, że $T_1 < T_2$. Jeżeli wartość gradientu w danym pikselu jest większa od T_2 , to zaznaczono ten punkt jako krawędź. Jeśli tak się stało, to zaczęto proces śledzenia krawędzi — dla każdego sąsiada, którego wartość gradientu jest większa od T_1 zaznaczono go jako krawędź. Jest ona wykonywana rekurencyjnie dla każdego zakwalifikowanego punktu.

Zamiast dokładnych wartości progowych można przekazać do funkcji 2 wartości — t_1, t_2 , które są procentem liczby pikseli, które będą niedopuszczone jako krawędzie. Dla $t_1 = 0.7, t_2 = 0.9$ dopuszczono tylko 10% pikseli jako te, które są większe od T_2 . Podając t_1, t_2 wyznaczono rozkład wartości gradientu w badanym obrazie, obliczono dystrybuantę $F(x)$ i wybrano dla T_1 ten argument, dla którego $F(x) = t_1 * \text{liczbapikseli}$ i analogicznie dla T_2 . W ten sposób wyznaczono progi do histerezy.

Operator Sobela [6] to metoda wyznaczania gradientu, a więc zarazem krawędzi zarówno w kierunku poziomym, jak i w pionowym. Dla każdego piksela przeprowadzono operację morfologiczną z następującymi maskami:

Maska rzędów			Maska kolumn		
-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

3.3. OPRACOWANY ALGORYTM PÓŁAUTOMATYCZNY

Po wykonaniu tych operacji otrzymano wartości s_1 i s_2 odpowiednio dla maski rzędów i kolumn. Na podstawie tych danych otrzymano następujące informacje o gradiencie:

Wielkość gradientu Kierunek krawędzi

$$\sqrt{s_1^2 + s_2^2} \qquad \tan^{-1} \left[\frac{s_1}{s_2} \right]$$

Detektory krawędzi oparte na gradiencie, w tym operator Canny’ego są często używane. Za ich główne zalety na podstawie [4] są:

- Dają dobre wyniki dla obrazów o dobrej jakości i bez szumów.
- Są wydajne - ich złożoność jest liniowa względem liczby przetwarzanych pikseli.
- Nie wymagają skomplikowanej sztucznej inteligencji do działania.

Zgodnie z [4] za ich główne wady można uznać:

- „Konieczność określenia rozmiaru maski i wartości progowej. Rozmiar maski znacząco wpływa na położenie miejsc, w których gradient przecina zera lub osiąga wartości maksymalne.
- Pomijanie narożników spowodowane faktem, że wartość 1D gradientu w narożnikach jest zwyczajnie mała.
- Operator pierwszej pochodnej wykrywa tylko schodkowe krawędzie.
- Duża wrażliwość na szum.”
- Na podstawie obserwacji działania algorytmu - rozmyte krawędzie często nie są wykrywane przez małe różnice w wartościach kolejnych sąsiadujących pikseli.

W ten sposób otrzymano macierz, gdzie każde pole w macierzy odpowiada pikselowi w wejściowej bitmapie — obrazie medycznym. Jeśli w komórce macierzy znajduje się 1, to w tym miejscu na bitmapie znajduje się krawędź, w przeciwnym przypadku 0. W ten sposób algorytm wykrył wszystkie znaczące krawędzie na bitmapie. Kolejnym krokiem przetwarzania było stworzenie grafu na podstawie wyżej wymienionej macierzy.

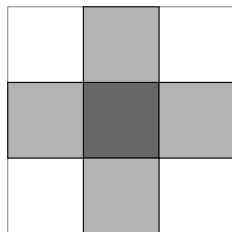
[11] [12]

3.3.2. Stworzenie grafu z bitmapy

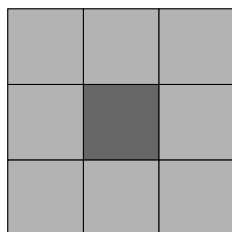
Na tym etapie algorytm potrzebuje następujących danych wejściowych:

- Macierz z wartościami logicznymi prawda/fałsz czy znajduje się danym punkcie krawędź. Może to być także realizowane poprzez macierz wartości liczbowych.
- Punkty wybrane przez użytkownika aplikacji.

Przed rozpoczęciem działania algorytmu należy zapewnić łączność 4-krotną (ang. Pixel 4-connectivity) [7]. Jest ona zwana także sąsiedztwem von Neumanna. Przy łączności 4-krotnej sprawdza się tylko sąsiadów w poziomie lub pionie.



Dla łączności 8-krotnej sprawdza się wszystkich możliwych sąsiadów, także po przekątnej. Jest ona zwana także sąsiedztwem Moore'a lub otoczeniem Moore'a [8].



W przypadku zastosowania łączności 8-krotnej przy wyznaczaniu długości krawędzi musiano by zastosować metrykę Czebyszewa, która jest specjalnym przypadkiem odległości Minkowskiego. Jeśli zostanie łączność 4-krotna to długość krawędzi byłaby obliczana zgodnie z metryką miejską, zwaną też metryką Manhattan.

Metryka Manhattan w kontekście dalszego przetwarzania w celu wyszukiwania najkrótszych ścieżek w grafie jest bardziej adekwatna, ponieważ jest intuicyjna w wyznaczaniu odległości na obrazie płaskim w porównaniu do metryki Czebyszewa. W tym przypadku najlepsza byłby tutaj metryka Euklidesa, lecz mamy do czynienia nie z kolejnymi punktami oddalonymi od siebie, a z sąsiadującymi pikselami. Ponadto w tym algorytmie istotne jest szybkie szacowanie odległości, czy też długości danej krawędzi.

Wykrywanie wierzchołków przy łączności 4-krotnej jest prostsze. Wystarczy zliczyć liczbę sąsiadów. Poniżej zakładamy, że piksel jest oznaczony jako krawędź w macierzy wejściowej. W zależności od liczby sąsiadów mamy następujące przypadki:

- 0 — wierzchołek izolowany,
- 1 — punkty końcowe (ang. endpixels),
- 2 — punkty łączące (ang. linkpixels), czyli fragmenty krawędzi,
- 3–4 — punkty węzłowe (ang. vertices), czyli punkty, od których odchodzą co najmniej 3 krawędzie.

W przypadku łączności 8-krotnej do detekcji wierzchołków należałoby stosować przekształcenia Hit-or-Miss z elementami strukturalnymi. Elementy strukturalne do wykrywania odpowiednich punktów są następujące:

- wierzchołek izolowany:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

- punkty końcowe (ang. endpixels):

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ z & z & z \end{bmatrix},$$

- punkty łączące (ang. linkpixels), czyli fragmenty krawędzi, posiadają dokładnie 2 sąsiadów,

- punkty węzłowe (ang. vertices), czyli punkty, od których odchodzą co najmniej 3 krawędzie:

$$\begin{bmatrix} z & 1 & z \\ z & 1 & z \\ z & z & 1 \end{bmatrix} \text{ lub } \begin{bmatrix} 1 & z & z \\ z & 1 & z \\ 1 & z & 1 \end{bmatrix},$$

Warto zauważyć, że te elementy strukturalne należy obracać o 90, 180, 270 stopni. Za każdym razem trzeba wielokrotnie sprawdzać te same piksele. Ponadto należy sprawdzać 8, a nie 4 sąsiadów.

Kolejnym problemem jest fakt, że przy spójności 8-krotnej przekształcenie Hit-or-Miss może w najbliższym otoczeniu punktu krzyżowania się krawędzi oznaczyć kilka otaczających punktów, jako punkty węzłowe. Jest to złe rozwiązanie, ponieważ w ten sposób może nawet kilkukrotnie zwiększyć liczbę wierzchołków w grafie, co przełożyłoby się na niską wydajność algorytmu.

Ostatnim problemem z jakim należałoby się wiązać wybierając łączność 8-krotną jest fakt, że macierz wejściową dla tego etapu algorytmu należałoby poddać procesowi szkieletyzacji. Najlepiej byłoby w tym celu skorzystać z algorytmu KMM [9] lub K3M [10]. Te algorytmy musiałyby co najmniej raz przejrzeć całą macierz z wykrytymi krawędziami w optymistycznym przypadku.

Z wyżej wymienionych powodów zdecydowano się na łączność 4-krotną. Przygotowano i zaimplementowano algorytm tworzący graf z bitmapy, a jego pseudokod znajduje się poniżej.

- 1) Dla każdego punktu oznaczonego jako krawędź w macierzy wejściowej
- 2) Jeśli punkt ma 1 sąsiada albo co najmniej 3 sąsiadów (tzn.
 jest albo punktem końcowym albo węzłowym)
- 3) Dodaj punkt jako nowy wierzchołek do grafu.

- 1) Dla każdego k w $\{ 0, 1, \dots, |V_2| - 1 \}$:
- 2) Dla każdej kombinacji k wierzchołków v ze zbioru V_2 :
- 3) Tworzymy graf G_3 , który powstaje poprzez usunięcie z G_2
- 4) wierzchołków należących do zbioru v (oraz przyległych krawędzi).
- 5) Ozn.: $G_3 = G_2 - x$.
- 6) Niech $n = |V_1| - |V_3|$.
- 7) Dla każdej kombinacji n wierzchołków x ze zbioru V_1 :
- 8) Dla każdej permutacji wierzchołków p w grafie $G_1 - x$:
- 9) Jeśli grafy G_3 oraz p są izomorficzne:
- 10) Zwróć G_3 jako wynik.

3.3.3. Zapewnienie spójności grafu

3.3.4. Wyszukanie najkrótszych ścieżek w grafie

3.3.5. Optymalizacja

3.4. Moduł obliczeń statystyk

4. Przeprowadzone eksperymenty

4.1. Zbiór testowy

4.2. Wydajność algorytmu półautomatycznego

4.3. Analiza wyników i wnioski

5. Podsumowanie

5.1. Napotkane problemy i ograniczenia

5.2. Możliwości dalszego rozwoju

Bibliografia

- [1] Nowacki R., Plechawska-Wójcik M.: Analiza porównawcza narzędzi do budowania aplikacji Single Page Application — AngularJS, ReactJS, Ember.js, *Journal of Computer Sciences Institute 2, Politechnika Lubelska, Instytut Informatyki*, Lublin, 2016, 98–103
- [2] Microsoft Corporation: Dokumentacja platformy .NET. Oficjalna strona: <https://docs.microsoft.com/pl-pl/dotnet/> [Dostęp 22 stycznia 2019]
- [3] Microsoft Corporation: Informacje o platformie .NET Core. <https://docs.microsoft.com/pl-pl/dotnet/core/about> [Dostęp 22 stycznia 2019]
- [4] Kronis K., Uhanova M.: Performance Comparision of Java EE and ASP.NET Core Technologies for Web API Developmnet. *Applied Computer Systems 23, Riga Technical University*, Ryga, 2018, 37–44
- [4] Cytowski J., Gielecki J., Gola A.: Cyfrowe przetwarzanie obrazów medycznych: Algorytmy. Technologie. Zastosowania. *Akademicka Oficyna Wydawnicza EXIT*, Warszawa, 2008, 88–94.
- [5] Canny J. F.: Finding Edges and Lines in Images. *Technical report no. 720, Massachusetts Institute of Technology (MIT)*, Cambridge, Massachusetts, USA, 1983.
- [6] Sobel I., Feldman G.: An 3x3 Isotropic Image Gradient Operator for Image Processing. *Presentation at Stanford Aartificial Intelligence Project (SAIL) in 1968*, 2014
- [7] Rosenfeld A., Kak A. C.: Digital Picture Processing, *Academic Press, Inc.*, Nowy Jork, 1982
- [8] Weisstein, Eric W.: Moore Neighborhood. *From MathWorld—A Wolfram Web Resource*. <http://mathworld.wolfram.com/MooreNeighborhood.html> [Dostęp 22 stycznia 2019]
- [9] Saeed K., Rybnik M., Tabędzki M., Adamski M.: Algorytm do Ścieniania Obrazów: Implementacja i Zastosowania *Zeszyty Naukowe Politechniki Białostockiej 2002 Informatyka - Zeszyt 1*, Białystok, 2002

- [10] Saeed K., Tabędzki M., Rybnik M., Adamski M.: K3M: A Universal Algorithm for Image Skeletonization and a Review of Thinning Techniques *International Journal of Applied Mathematics and Computer Science*, 2010, 20(2) Białystok, 2010, 317–335
- [11] Sedgewick R., Wayne K.: Algorytmy Wydanie IV, *Helion*, Gliwice, 2012, 526–706
- [12] Albahari J., Albahari B.: C 6.0 w pigułce, *Helion, O'Reilly Media, Inc.*, Gliwice, 2016, page–page

Instrukcja instalacji

Instrukcja użytkowania

Wykaz symboli i skrótów

nzw. nadzwyczajny

* operator gwiazdka

~ tylda

Jak nie występują, usunąć.

Spis zawartości załączonej płyty CD

Spis załączników

1. Załącznik 1
2. Załącznik 2
3. Jak nie występują, usunąć rozdział.